Exploring the Design Space of Symbolic Music Genre Classification Using Data Mining Techniques

Ortiz-Arroyo, Daniel; Kofod, Christian

Published in:
International Conference on Computational Intelligence for Modeling Control and Automation

Publication date:
2008

Document Version
Early version, also known as pre-print

Exploring the Design Space of Symbolic Music Genre Classification Using Data Mining Techniques

Christian Kofod and Daniel Ortiz-Arroyo
Electronics Department
Aalborg University
Niels Bohrs Vej 8,
6700 Esbjerg, Denmark

Abstract

This paper describes a method based on data mining techniques to classify MIDI music files into music genres. Our method relies on extracting high level symbolic features from MIDI files. We explore the effect of combining several data mining preprocessing stages to reduce data processing complexity and classification execution time. Additionally, we employ a variety of probabilistic classifiers and ensembles. We compare the results produced by our best classifier with those obtained by more complex state of the art classifiers. Our experimental results indicate that our system constructed with the best performing combination of data mining preprocessing components together with a Naive Bayes-based classifier is capable of outperforming other more complex ensembles of classifiers.

1 Introduction

Some music genre classification systems emulate the way humans proceed to perform this task. When asked to classify music we are commonly provided with a list of representative titles of the genre. One is then expected to gain an understanding of the genre by generalizing from the combination of properties that characterize these given titles. Classification of new music titles is performed by evaluating their similarity with respect to the other titles that we already know belong to a certain category. This is one feature of the *inductive* learning process and one example of the kind of problem that classification algorithms are designed to solve.

In this paper we use an empirical approach aimed at finding the best performing classifier for symbolic music genre classification. The media format employed as input to our classification system is *symbolic* audio in the form of standard General MIDI (GM) files. Contrarily to real audio samples, MIDI files contain information on actual musical events such as note-on and note-off events, tempo and meter-changes, etc. that is not available in other formats like WAVE or MP3. Using this information it is possible to extract high-level musical features such as the fraction of notes played by a certain instrument, the amount of tri-tones in a recording, etc. In this work we use exclusively these musical properties to classify genre, following the definition of van der Merwe [11, p. 3]: *"A music genre is a category (or genre) of pieces of music that share a certain style or 'basic musical language' "*.

Our classification system extracts 1024 high-level musical features from the MIDI files and selects the most representative using a correlation-based feature selection mechanism. The method employed utilizes a best-first search approach and heuristics to maximize feature-to-class correlation, while minimizing at same time inter-feature correlation. Afterward, the data is discretized using a method based on the minimum description length principle (MDLP) and information theory . Finally training and classification are performed with a variety of classifiers. We used the Weka data mining experimentation environment to explore the design space of our classification system, employing diverse combinations of preprocessing steps. Finally, our classification system is evaluated with a 10 times 10-fold cross-validation. The experimental results obtained show that our best performing classifier is capable of outperforming other more complex hierarchical classifiers and is comparatively simpler in structure.

The paper is organized as follows. Section 2 contains a summary of the most relevant related work. A brief description of the proposed methods is presented in Section 3, followed by the experimental results we obtained in Section 4. Finally, Section 5 contains a number of conclusions and describes future work.

978-0-7695-3514-2/08 $25.00 © 2008 IEEE
DOI 10.1109/CIMCA.2008.223

2 Related work

Classification on real-audio music has been reported elsewhere e.g. [3], [2], [14], [17]. In this paper we present a summary of previous research in symbolic music classification.

Basili et al. describe in [1] some experiments with 300 MIDI files with the Humdrum[1] toolkit and Weka. Five algorithms are evaluated: Naive Bayes, VFI (Voting Features Interval), J48/PART[2], NNge (Nearest Neighbor using untested generalized exemplars), and JRip (A rule-based classifier implementing a propositional rule learner). Recordings belong to one of 6 major genres: Classical, jazz, rock, blues, disco, and pop. Extracted features are purposely limited to few and relatively easily extracted ones such as melodic intervals, instrument classes, and time and meter changes.

Both split- and cross-validation are used for evaluating multi-class and binary classification. In their experiments J48 performs well, when compared to other methods, obtaining a cross-validation accuracy results of approximately 60%. In line with our findings presented in this paper, naive Bayes outperforms all methods with an improvement of around 10% over the second-best method.

Another interesting approach is taken by Ruppin & Yeshurun, [13], who look at repeating patterns in music that may be used in the classification process. Working on monophonic MIDI melody lines, they show the effectiveness of using a distance similarity measure built using compression techniques to compare between melody lines, using the comparison result as a feature for classification. Their method takes into account four recurring musical transformations: Transposition (global pitch change), augmentation/diminution (global tempo change), sequential modulation (parts played at different pitch) and crab transformation (inversion of pitch). Their method, in brief, is to remove all MIDI messages except note-on events, and then remove the mentioned transformations. k-nearest neighbor is applied to the compression distances calculated with LZW compression [16]. Results on 50 MIDIs and three genres (classical, pop, and traditional Japanese music) are promising with an 85% genre match and a 58% composer match. Among their conclusions, they find that repetition occurs very often in music, and that this fact can be exploited for classification.

McKay in [9] employs a number of hierarchical classifier ensembles. His system, called *Bodhidharma* relies on an array of 111 high-level features, ten of which are *multi-dimensional*. Contrarily to single dimension features, multi-dimensional features have a number of associated sub-values. The program accepts user-defined genre taxonomies, and is tested not only with a 9 genre dataset but also with a larger hierarchically organized dataset containing 38 leaf genres in three levels. The program can assign multiple genres to one recording, and also determine the degree to which it belongs to these genres. The base classifiers employed are k-nearest neighbor, neural networks, and genetic algorithms.

The extracted high-level features belong to the groups: *Melody, chords, pitch, dynamics, rhythm, texture,* and *instrumentation*. To process the multi-dimensional features, for each branch in the genre taxonomy, three classifier ensembles are trained: 1) one parent ensemble that deals with direct descendants of the current node in the taxonomy, 2) one flat, leaf ensemble that classifies all leaf categories in the current branch, and 3) one flat classifier that classifies each pair of leaf categories. The ensembles are structurally identical and work by taking in the complete set of features and outputting a non-normalized score in the unit-interval for each candidate category. The ensembles are comprised of one k-nearest neighbor classifier that takes as input the one-dimensional features, and one neural network-based classifier for each of the multi-dimensional features. The final score of each ensemble is a weighted average of the outputs of the internal classifiers with weights optimized by genetic algorithms.

This paper describes an empirical approach aimed at finding the best performing data mining preprocessing steps and classifier that produce the highest accuracy in classifying music genre using symbolic information. The approach presented in this paper has some similarities with two of the methods mentioned above. Like Basili et al. we employ a single and relatively simple base classifier and as in McKay's work, we use a multitude of high-level features and ensembles of classifiers. However, in contrast with both approaches, we also apply some data mining preprocessing steps that help to reduce processing complexity on the data input. In our experiments we used the same data sets employed in [9] e.g. CM-38 and CM-9. This enables us to compare our results against those presented in [9], which presents the classifier that has shown the best performance results reported so far in the literature.

The goal of this work is to explore the design space of a classification system for symbolic audio using data mining techniques and probabilistic classifiers. For comparison purposes we also present the effect of our special settings-combination on J48 induced trees. We used decision trees as they have the advantage of producing a relatively more readable and easy to understand classification representation for the non specialist, in spite of generally achieving lower classification accuracy when compared to other methods.

[1] See `http://music-cog.ohio-state.edu/Humdrum/`.

[2] J48 is the WEKA equivalent of C4.5. PART is a rule extractor for J48

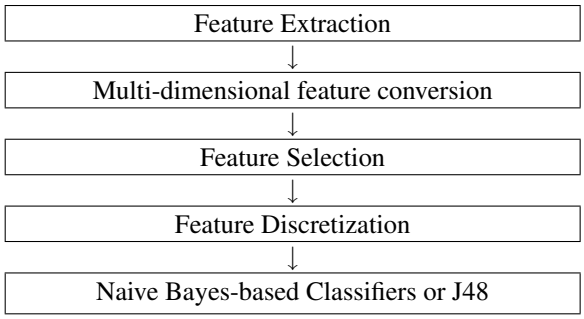

Figure 1. Basic classification learning processing

3 Description of the Proposed Method

The combination and application order of algorithms comprised by our proposed classification method is depicted in Figure 1:

As is illustrated in previous figure, the processes of feature extraction and selection and are commonly employed in data mining tasks. However, exploring its effect and showing the benefit of its application in the domain of symbolic music genre classification is one of the main contributions of this paper.

Our classification system first extracts a total of 111 features from a set of training recordings using a software component called JSymbolic [10]. JSymbolic is capable of extracting multi-dimensional features from midi files. The features extracted belong to the following categories: instrumentation (type of instrument), texture (number of voices and its interaction), rhythm (meters and rhythmic patterns), dynamics (the dynamic range), pitch statistics (occurrence rates of notes), melody (melodic intervals and variations), and chords (types of chords). A detailed discussion of all features that JSymbolic is capable of extracting is provided in [9, pages 55–76].

The use of multi-dimensional features has some advantages in the context of a multi-classifier system like Bodhidharma, since the hierarchy of classifiers can be used efficiently to processes features and its sub-features. However, since the classifiers used in our experiments do not support multi-dimensional features directly, ten of the multi-dimensional features extracted by J Symbolic are *flattened* first.

To *flatten* multi-dimensional features, each of their sub-features is first promoted into independent, one-valued features. This processing produces a total of 1024 one-dimensional features. The resulting features are then passed to CfsSubset[6][5], which is a filtering-type feature selection mechanism. CfsSubset basic goal is to try to improve accuracy (by removing features that are highly correlated to other features), and reduce complexity (by reducing the number of features). This automated feature selection method uses a best-first type search together with a correlation-based quality measure. CfsSubset basically selects features with as little feature-to-feature correlation and as much feature-to-class correlation as possible. The resulting filtered features are then *discretized* to convert their numeric values into discrete ranges of values. The discretization step is performed with a method based on the Minimum Description Length Principle(MDLP) as is described in [4]. The MDL principle was originally proposed to perform inductive inference by looking at regularities in the data that caould be used to compress it. MDLP principle together with information theory is used in data discretization [4] to estimate the cost of deciding when to partition or not the data. Finally, the produced set of flattened, selected, and discretized features are then passed on to the classifiers.

The *Naive Bayes* (NB) classifier is one of the simplest probabilistic classification systems available. The NB model assumes complete independence between the random variables that represent the attributes employed. One advantage of using the independence assumption is that training is simplified as there is no need to calculate the whole joint probability distribution. In spite of using this strong simplifying assumption, Naive Bayes has shown to perform well in many domains. Another classifier is *Hidden Naive Bayes* (HNB)[18], which is an extension of NB that relaxes the strong independence assumption employed by NB. HNB works by assigning an extra layer of so-called *hidden* nodes to the pre-defined Naive Bayes network, so that each attribute node is the child of the class node and of one such hidden node. Each of the hidden nodes are designed to represent the effect of the surrounding network structure on the attribute at hand, thus allowing the remaining network to affect the attribute node without having to actually model these dependencies.

Average One-Dependence Estimator (AODE) is another classifier based on NB [15] that allows each of the attribute nodes to be dependent on at most one other attribute node. Given that each feature must depend on one other feature each, a form of model-selection must take place. In AODE, this is performed by using an aggregate of one-dependence classifiers. The final prediction is made by averaging the predictions of these classifiers. *Weightily Average One-Dependence Estimator* (WAODE) [7] is an extension to AODE that enforces a weight value for each attribute depending on its correlation with the class label.

Ensembles of classifiers can be constructed using some of the previously discussed base classifiers together with some form of voting or weighting mechanism. Bagging is one method that works on ensembles by manipulating the input data for a predefined number of same type base-

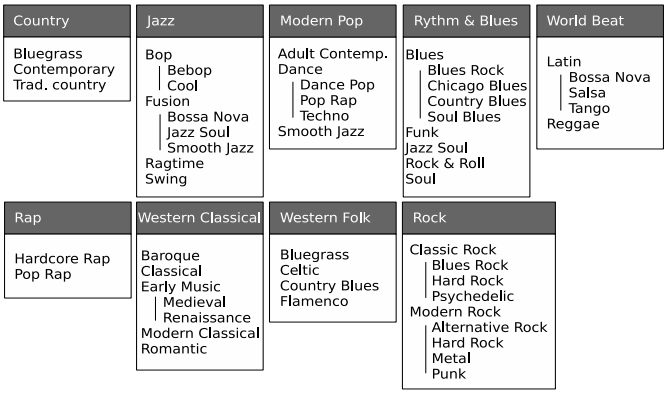

Figure 2. The CM-38 genre taxonomy.

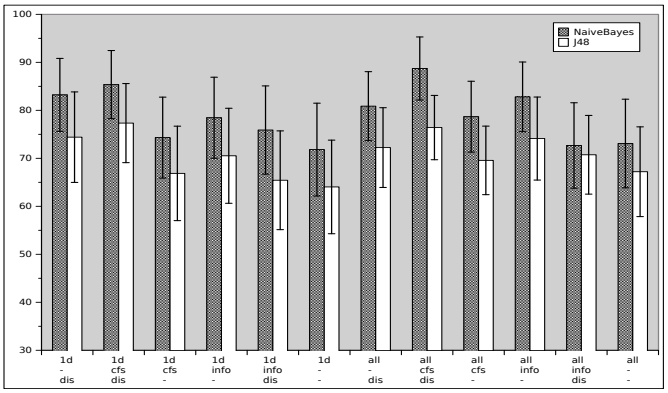

Figure 3. Experimental results for CM-9

learners in order to create variance among them. Bagging, short for *bootstrap aggregation*, creates its datasets from the original training dataset by sampling with replacement from it and training each learner on one of the resulting datasets. Once trained, the ensemble is used for classification by running the new instance through each classifier and combining their results by means of voting [8].

4 Experimental Methodology and Results

To explore the design space of our classification system, a series of experiments were performed on different datasets. We use different combinations of data mining techniques and classifiers. In our experiments we employed single classifiers such as NB and HNB additionally to J48 decision trees and ensembles of classifiers.

Experimental evaluations were performed using 10 times 10-fold stratified cross-validation. Using 10-fold cross-validation the data set is divided randomly into 10 sets, 9 sets are used for training and one set for testing. The process is repeated 10 times changing the training and test sets

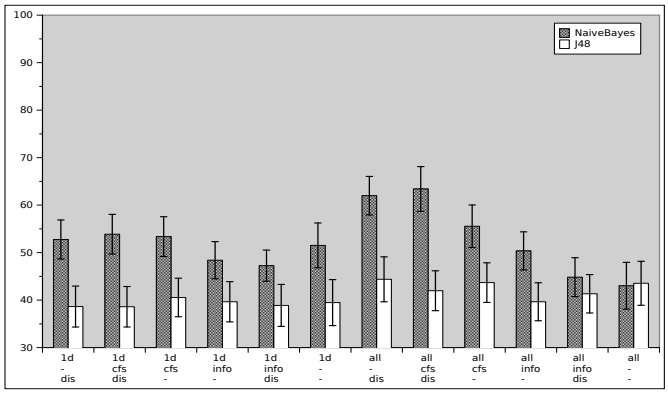

Figure 4. Experimental results for CM-38

every time, averaging the results from each experiment and calculating the standard deviation for all the runs.

The datasets denoted as CM-9, and CM-38 were used in the evaluation. CM-9 and CM-38 were created by McKay in [9] under the names of T-9 and T-38. The former consists of 225 recordings with 25 recordings in 9 slightly more specialized genres: Bebop, jazz-soul, swing, rap, punk, country, baroque, modern-classical, and romantic-classical. CM-38 has 950 recordings with 25 recordings and 38 leaf genres arranged in three levels as depicted in figure 4. The inclusion of CM-9 and CM-38 facilitates direct comparison with the state of art classifier that has reported the best performance results so far in [9].

Experimental results on the effect of a diversity of settings used are given for datasets CM-9 and CM-38 in figures 3 and 4 respectively. As classifiers we have used Naive Bayes and J48.

Results are given in terms of the average classification accuracy obtained with different combinations of settings over each of the datasets. Labels for the settings (axis X) have the following meaning:

all: Classification using all 1024 features.

1d: Classification using only the 101 one-dimensional features.

cfs: Features were subjected to CFSSubset feature selection algorithm.

info: Features were ranked with info-gain and only the top-30 features were used for classification.

dis: Features were discretized with the MDL-based discretization algorithm.

Figures 3 and 4, show that the combination of data mining preprocessing steps that consistently provides the best performance using a Naive Bayes classifier, consists of all 1024 flattened, high-level features together with a

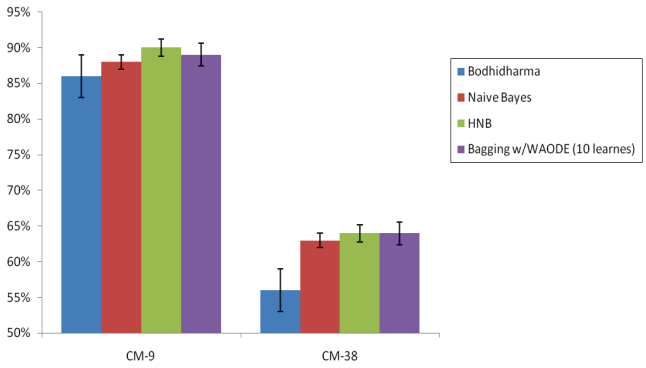

Figure 5. Classification accuracies for CM-9 and CM-38 with Bodhidharma and the proposed method using diverse classifiers.

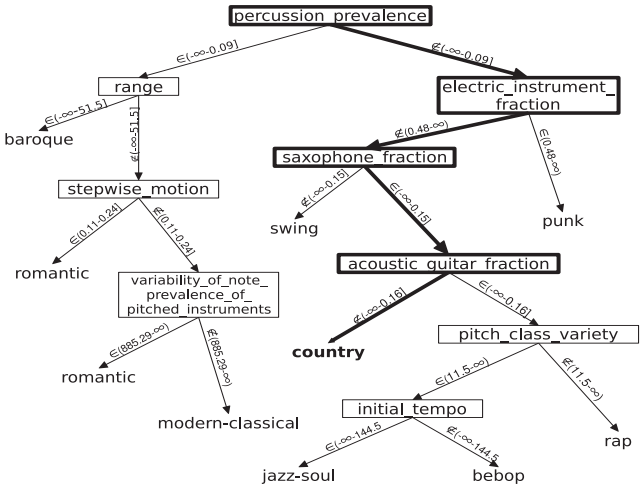

Figure 6. Example of a size-optimized tree

CfsSubset-based feature selection and MDLP-based discretization of numerical values. These results also show that the data mining preprocessing stages when a J48 decision tree is used as classifier provide different performance results that depend on the data set used.

Once we determined the best combination of data mining preprocessing steps we perfomed a comparison of results with McKay's Bodhidharma and the proposed method on datasets CM-9 and CM-38. Results are given in figure 5 in terms of overall averaged accuracy.

In our experiments we used a wide variety of classification methods ranging from a single Naive Bayes classifier, HNB, AODE, and WAODE together with a diversity of ensembles of Naive Bayes-based classifiers using techniques such as standard voting mechanisms (e.g. majority, Borda, Condorcet), Bagging and Boosting (MultiBoost and AdaBoost), additionally to Bayesian Networks and Sphere Oracle[12]. As some of these methods were not available in Weka we had to implement them to asses their performance. However, for lack of space we report exclusively the results obtained by the classification methods that showed the best performance in all our experiments. These methods were Naive Bayes, HNB, and an ensemble of 10 WAODE base classifiers using Bagging.

McKay has reported the best results known so far on symbolic audio using his Bodhidharma system with an 86% overall accuracy on a 9 category taxonomy (CM-9), and 57% on the more elaborate 38 leaf genre taxonomy (CM-38). As for the system's execution performance, McKay in [9] reports a computation time for one-fold out of a 5-fold cross-validation session of approximately 89 minutes.

Figure 5 shows that HNB achieves the best performance among the single classifiers together with Bagging with an ensemble of 10 WAODE classifiers. HNB achieves an average of 90% of accuracy on the CM-9 data set and 64% on the CM-38 data set. In comparison an ensemble of 10 WAODE classifiers using Bagging achieves 89% of acuracy on CM-9 and 62% on CM-38 datasets. These results show that our classification system outperforms Bodhidharma by 4-3% on average on CM-9 and 7-5% on CM-38 data set, respectively. The standard deviation shown by our system is smaller, due to the fact that we used 10-fold cross validation. In comparison, McKay used 5-fold cross validation.

Regarding training time, our method achieves an execution time of under 1 minute using a 10-fold cross-validation session (including feature-selection, discretization, training and evaluation) in the same datasets used by McKay to evaluate Bodhidharma.

We also experimented applying our method to a dataset similar to CM-9 but with four times as many training samples, and a less specialized genre taxonomy. However, performance was not improved over the highest we have obtained.

Finally, we experimented with J48's generated decision trees. We fine tuned this induction method to produce the smallest possible trees with the idea of improving its readability, while maintaining at same time an acceptable accuracy. The technique consisted of increasing *pruning confidence* value, enforcing the use of *binary splits*, and increasing the minimum number of *instances per leaf*. The average decrease in the number of leaf nodes obtained by the produced trees when using these settings was 76% with an average decrease in accuracy of 4.06%. An example of a size-optimized tree produced for dataset CM-9 (selected and discretized) is shown in figure 6. This particular tree has 10 leaf-nodes and 19 branches. Using the default J48 settings, the same tree has 33 leaf-nodes and 54 branches.

5 Conclusions and Future Work

The combined use of 1024 flattened, high-level features, the CfsSubset-based feature selection, the MDL-based discretization of numerical values, and probabilistic classifiers based on extensions to Naive Bayes have been shown to significantly outperform the best results reported so far in the literature [9]. Our results also indicate that probabilistic classifiers based on either using ensembles of WAODE learners or a single Hidden Naive Bayes classifier are more appropriate for the task.

The improvements in accuracy obtained by our classification system have the additional benefit of having lower execution time. Our system was able to perform the classification in the range of 41-45 seconds using the most accurate classifiers. This execution time includes the process of selection, discretization, training and classification. Comparatively [9] reports a 96 hour training period on the same CM-9 dataset due to the use of a hierarchical system of artificial neural networks and optimizing genetic algorithms.

Our comprehensive set of experiments based on probabilistic classifiers indicates that the problem of symbolic music genre classification may be reaching a limit in the accuracy provided by the current classification methods we have available to date. Our experiments also show that using the current methods based on ensembles of classifiers does not improve classification accuracy. In future work we plan to apply a similar classification approach to real audio music. However, as the number of high level features that can be extracted from real audio is much more limited we will concentrate our efforts on improving the accuracy of the base classifier.

6 Acknowledgments

The authors would like to thank especially Cory McKay, from McGill University, Canada, for supplying two of his MIDI repositories and the jSymbolic feature extractor.

References

[1] R. Basili, A. Serafini, and A. Stellato. Classification Of Musical Genre: A Machine Learning Approach. *ISMIR 2004: 5th International Conference on Music Information Retrieval*, 2004.

[2] J. J. Burred and A. Lerch. A Hierarchical Approach To Automatic Musical Genre Classification. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*, Sept. 2003.

[3] R. B. Dannenberg, B. Thom, and D. Watson. A machine learning approach to musical style recognition. In *In Proceedings of the 1997 International Computer Music Conference*, pages 344–347. International Computer Music Association., 1997.

[4] U. M. Fayyad and K. B. Irani. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In *IJCAI*, pages 1022–1029, 1993.

[5] M. A. Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, Waikato University, 1998.

[6] M. A. Hall and L. A. Smith. Feature Subset Selection: A Correlation Based Filter Approach. In *International Conference on Neural Information Processing and Intelligent Information Systems*, pages 855–858. Springer, 1997.

[7] L. Jiang and H. Zhang. Weightily Averaged One-Dependence Estimators. In Q. Yang and G. I. Webb, editors, *PRICAI 2006: Trends in Artificial Intelligence, 9th Pacific Rim International Conference on Artificial Intelligence*, volume 4099 of *Lecture Notes in Computer Science*, pages 970–974. Springer, 2006.

[8] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[9] C. McKay. Automatic Genre Classification of MIDI Recordings. Master's thesis, McGill University, Montreal, June 2004.

[10] C. McKay and I. Fujinaga. jSymbolic: A feature extractor for MIDI files. In *Proceedings of the International Computer Music Conference*, 2006.

[11] P. V. d. Merwe. *Origins of the popular style : the antecedents of twentieth-century popular music*. Clarendon, 1989.

[12] J. J. Rodríguez and L. I. Kuncheva. Naive Bayes Ensembles with a Random Oracle. In *Multiple Classifier Systems, 7th International Workshop, MCS 2007*, volume 4472 of *Lecture Notes in Computer Science*, pages 450–458. Springer, 2007.

[13] A. Ruppin and H. Yeshurun. MIDI Music Genre Classification by Invariant Features. In *ISMIR 2006, 7th International Conference on Music Information Retrieval*, pages 397–399, Oct. 2006.

[14] G. Tzanetakis, G. Essl, and P. Cook. Automatic Musical Genre Classification of Audio Signals. In *ISMIR 2001, 2nd International Symposium on Music Information Retrieval*, Oct. 2001.

[15] G. I. Webb, J. R. Boughton, and Z. Wang. *Not so naive Bayes: aggregating one-dependence estimators*, volume 58. Kluwer Academic Publishers, 2005.

[16] T. A. Welch. A Technique for High-Performance Data Compression. *IEEE Computer*, pages 8–19, June 1984.

[17] Y. Yaslan and Z. Cataltepe. Audio Music Genre Classification Using Different Classifiers and Feature Selection Methods. *The 18th International Conference on Pattern Recognition (ICPR'06)*, 2006.

[18] H. Zhang, L. Jiang, and J. Su. Hidden Naive Bayes. In M. M. Veloso and S. Kambhampati, editors, *The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference*, pages 919–924. AAAI Press / The MIT Press, 2005.